UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/437,169	11/10/1999	MARK D. RUSTAD	977.029US1	7612

21186 7590 01/15/2003

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER
KING, JUSTIN

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 01/15/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/437,169	RUSTAD, MARK D.	
	Examiner	Art Unit	
	Justin I. King	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the corr spondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 October 2002* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5, 8, 11-20, 23-38, 41-46, 49, 52, 55-64, and 67-77* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,8,11-20,23-38,41-46,49,52,55-64 and 67-77* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Drawings

1. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the cache memory reset (as in the claim 1) must be shown or the feature(s) canceled from the claim(s). No new matter should be entered. The drawing does not show whether the present processor's cache memory or the previous processor's cache is been reset. The draw needs to show the claimed feature and has to be supported by the specification's disclosure.

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 32-33 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 32 recites the limitation "the present processor" in claim 32's lines 2-3. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 30-32 and 34-35 are rejected under 35 U.S.C. 102(b) as being anticipated by Gusefski et al. (U.S. Patent No. 5,202,972).

Referring to claim 30: Gusefski discloses a machine-readable medium (figure 2, structure 26's L2 cache) in a storage controller/bus switch unit (figure 2, structure 26). In order to properly store and retrieve the data, the memory has a particular file system structure; such that, the stored software data source is a data structure. Gusefski discloses a plurality of processors, and at least one processor has a fast memory (figure 2).

Gusefski's storage controller/bus switch unit controls the access to the shared source (figure 2, structures 102, 106, column 6, lines 7-20), and Gusefski discloses that bus switch unit only grants access exclusively to one of the units at a time (column 6, lines 11-12); thus, Gusefski's bus switch unit's operation on only granting one access at a time is equivalent to the state for indicating the resource is under control.

Gusefski discloses the data structure includes an act for resetting the processors' caches. Gusefski discloses that a processor 20c modifies the data on shared memory and other processors also have their caches mapped to that data, the processor 20c cannot re-access the shared resource until other processors update their caches (column 6, lines 26-42). Thus, Gusefski's

operation on prevention processor 20c to re-access the shared resource is equivalent to the identifier for a past processor. Hence, the claim 30 is anticipated by the Gusefski.

Referring to claim 31: Claim 30's argument applies; furthermore, a group of data structure is a class and Gusefski's data structure includes a group of instructions (column 6, lines 12-15). Gusefski further discloses the data structure includes an act for resetting the present processors' caches (column 6, lines 26-42).

Referring to claim 32: Claims 30-31's arguments apply; furthermore, Gusefski's bus switch unit controls and grants an exclusive access to the shared resource (column 6, lines 11-12); thus, Gusefski's operation on granting access to one particular processor is equivalent to the second identifier for identifying the present processor that has exclusive control of the resource.

Referring to claim 34: Claim 30's argument applies; furthermore, Gusefski's fast memory is a cache memory (figure 2, structure 18).

Referring to claim 35: Claim 30's argument applies, furthermore, Gusefski discloses the data structure including a type of data that is adapted to represent at least one portion of the resource (column 6, lines 24-25).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

8. Claims 1-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange et al. (U.S. Patent No. 3,845,474) in view of Averill (U.S. Patent No. 5,313,591).

Referring to claim 1: Lange discloses a system including a bus, a resource, and a plurality of entities (figure 1), and at least one entity among the plurality of entities including a memory (figure 1, structures 11).

Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity; however, Lange resets entity's memory upon gaining the resource control (column 2, lines 7-10, 45-49), and Lange's memory resetting operation at the time of the gaining the resource control rather than at the time of giving away the resource control is the selective reset.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control rather than at the time of gaining the resource control is the selective reset.

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the

multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control.

Referring to claim 2: The claim 2 is rejected over Lange in view of Averill as stated above; furthermore, Lange's entity is an integrated circuit (column 4, lines 2-8).

Referring to claim 3: The claim 3 is rejected over Lange in view of Averill as stated above; furthermore, Lange's resource is a main storage, which is a memory device.

Referring to claim 4: The claim 4 is rejected over Lange in view of Averill as stated above; furthermore, Lange discloses a manager (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to manage the access to the resource.

Referring to claim 5: The claim 5 is rejected over Lange in view of Averill as stated above; furthermore, Averill discloses that an arbitration protocol to arbitrate the shared bus among processors (column 1, lines 17-22), and teaches one on how to arbitrate the shared bus (column 2, lines 40-52). Averill's arbitration protocol is equivalent to the arbiter. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Averill's teaching into Lange because Averill teaches one on how to arbitrate the shared bus when multiple processors simultaneously contend for control.

9. Claims 8, 11, and 14-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski.

Referring to claim 8: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11).

Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity; however, Lange resets entity's memory upon gaining the resource control (column 2, lines 7-10, 45-49), and Lange's memory resetting operation at the time of the gaining the resource control rather than at the time of giving away the resource control is the selective reset. Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control rather than at the time of gaining the resource control is the selective reset

Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource, but Lange does not disclose the switch mechanism coupled to a central computing unit. Gusefski teaches the switch mechanism coupled to a central computing unit (figure 2, structure 28).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the teachings of Averill and Gusefski into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control and Gusefski teaches one to couple a central computing unit functioning as a shared channel processor to handle channel storage requests.

Referring to claim 11: Claim 8's argument applies; furthermore, Gusefski's switch mechanism is a hardware device.

Referring to claims 14-15: Claim 8's argument applies; furthermore, Lange's shared resource is a memory, which is a hardware device.

Referring to claim 16: Claim 8's argument applies; furthermore, the central computing unit coupled to Gusefski's switching mechanism is a communication channel controller.

Referring to claim 17: Claim 8's argument applies; furthermore, the data stored in Lange's shared resource is the software resource.

Referring to claim 18: Claims 8 and 17's arguments apply; furthermore, in order to properly store and retrieve the data, the memory has a particular file system structure, such as the FAT or FAT16, such that, the stored software data source is a data structure.

Referring to claim 19: Claim 8's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is a cache memory.

10. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski and the Structured Computer Organization by Andrew Tanenbaum.

Referring to claim 12: Claim 8's argument applies; furthermore, none of Lange, Averill, or Gusefski discloses a software switch. Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange, Averill, and Gusefski because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

11. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski, Tanenbaum and Georgiou et al. (U.S. Patent No. 4,633,394).

Referring to claim 13: Claims 8 and 12's arguments apply; furthermore, none of Lange, Averill, Gusefski, or Tanenbaum discloses that the switch is a Djikstra primitive. Georgiou teaches implementing distributed arbitration among multiple processors and shared resource with Djikstra primitive. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Georgiou's teaching into Lange, Averill, Gusefski, and Tanenbaum because Georgiou teaches one to route data with the Djikstra primitive to achieve the shortest path from the source to the destination.

12. Claims 20 and 26-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski.

Referring to claim 20: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11).

Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity; however, Lange resets entity's memory upon gaining the resource control (column 2, lines 7-10, 45-49), and Lange's memory resetting operation at the time of the gaining the resource control rather than at the time of giving away the resource control is the selective reset. Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-

40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control rather than at the time of gaining the resource control is the selective reset

Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource, but Lange does not disclose the switch mechanism coupled to a central computing unit and any lock means. Gusefski teaches a storage controller/bus switching unit (figure 2, structure 26) coupled to a central computing unit (figure 2, structure 28). Gusefski further discloses it is known to lock the shared resource while one of the processors is using the resource (column 1, lines 60-64). Hence, Gusefski's storage controller is equivalent to Applicant's lock. Therefore, it would have been obvious to one having ordinary skill in the computer art to adapt the teachings of Averill and Gusefski into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control and Gusefski teaches one to couple a central computing unit functioning as a shared channel processor to handle the channel storage request.

Referring to claim 26: Claim 20's argument applies; furthermore, the Gusefski's bus switching unit and the shared channel processor are the communication channel controller.

Referring to claim 27: Claim 20's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is cache memory.

Referring to claim 28: Claims 20 and 27's argument applies; furthermore, Lange does not specify the cache memory as a primary cache (L1 cache); Gusefski teaches that it is known to use L1 cache (figure 2, structure 18).

Referring to claim 29: Claim 20's argument applies; furthermore, Lange discloses that entity has a cache, although Lange does not specify particular types of cache, such as a secondary cache (L2 cache); every cache type is functioning at the same purpose as to store local information for speedy processing; the type of cache in which each processor uses is inconsequential for the invention as a whole and presents no new or unexpected results, so long as the data on the cache is reset.

13. Claims 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski and Tanenbaum.

Referring to claims 23-24: Claim 20's argument applies; furthermore, Gusefski does not specify whether the lock is a hardware register or a software semaphore. Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange, Averill, and Gusefski because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

Referring to claim 25: Claims 20 and 24's arguments apply; furthermore, a binary object is generated after the software source code is compiled.

14. Claims 33 and 36 and are rejected under 35 U.S.C. 103(a) as being unpatentable over Gusefski in view of Lange.

Referring to claim 33: Gusefski's disclosure is stated in the 102(b) Rejection above, but Gusefski does not explicitly disclose resetting the present processor's fast memory when the first identifier for the previous processor is different from the second identifier for the present processor. Lange teaches that it is know to reset the present processor's fast memory upon gaining the control on the shared resource (column 2, lines 7-10). Hence, it would have been obvious to one having ordinary skill in the computer art to modify Gusefski with Lange's teaching because Lange teaches one to resolve the memory coherency problem by resetting cache memory every time when the associated processor gains the control on the shared resource.

Referring to claim 36: Gusefski's disclosure is stated in the 102(b) Rejection above; furthermore, Gusefski discloses that the data structure includes at least one dimension of at least one portion of the resource (column 6, lines 24-25), however Gusefski does not explicitly disclose a list including at least one location of at least one portion of the resource. Lange teaches that the data structure on cache including at least one location of at least one portion of the resource (figure 3). Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Lange's teaching into Gusefski because Lange teaches one on how to utilize the cache memory for storing particular data type.

15. Claims 37-38 and 41are rejected under 35 U.S.C. 103(a) as being unpatentable over Gusefski in view of Lange.

Referring to claim 37: Gusefski discloses a plurality of processors and shared resources (figure 2), an exclusive control over the shared resource to only one processor (column 6, lines 11-12), and the processor has a fast memory (figure 2, structure 18).

Gusefski discloses that a processor 20c modifies the data on shared memory and other processors also have their caches mapped to that data; the processor 20c then cannot re-access the shared resource until other processors update their caches (column 6, lines 26-42). Thus, Gusefski's operation on prevention processor 20c to re-access the shared resource is equivalent to the identifier for a past processor.

Gusefski discloses a bus switch unit for distributing the exclusive control over the shared resource to one of the processor (column 6, lines 11-12); thus, Gusefski's operation on selecting processor for distributing the shared resource is equivalent to the identifying a present processor for exclusive control over the resource as a second identity. Gusefski discloses a comparing means (column 6, lines 35-42).

Gusefski does not disclose resetting the present processor's fast memory when the past processor is different from the present processor. Lange teaches that it is known to reset the present processor's fast memory upon gaining the resource control (column 2, lines 8-9).

Hence, it would have been obvious to one having ordinary skill in the computer art to modify Gusefski with Lange's teaching because Gusefski teaches one on how to resolve the cache coherence problem.

Referring to claim 38: Claim 37's argument applies; further, Gusefski's fast memory is a cache memory.

Referring to claim 41: Claim 37's argument applies; furthermore, the resource distribution must be done according to a particular orderly fashion, and Gusefski's bus switch unit decides each request's priority (column 5, lines 38-39); the FIFO is a basic algorithm for handling transaction request.

16. Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gusefski in view of Lange.

Gusefski discloses a plurality of processors and shared resources (figure 2) and the processor has a cache memory (figure 2, structure 18). Gusefski further discloses an exclusive control over the shared resource to only one of the requesting processors (column 6, lines 11-12), and a bus switch unit for scheduling the resource access (column 5, lines 38-40).

Gusefski does not disclose resetting the current processor's cache memory when the current processor is different from the previous processor. Lange teaches that it is known to reset the present processor's fast memory upon gaining the resource control (column 2, lines 8-9).

Hence, it would have been obvious to one having ordinary skill in the computer art to modify Gusefski with Lange's teaching because Lange teaches one on how to resolve the cache coherence problem.

17. Claims 43-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski.

Referring to claim 43: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11).

Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity; however, Lange resets entity's memory upon gaining the resource control (column 2, lines 7-10, 45-49). Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40).

Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource, but Lange does not disclose the switch mechanism coupled to a lock. Gusefski teaches the switch including a lock means for granting exclusive access to only one processor at a time (column 6, lines 11-12). Gusefski discloses a central computing unit for channel management (figure 2, structure 28).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the teachings of Averill and Gusefski into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control and Gusefski teaches one to couple a central computing unit functioning as a shared channel processor to handle channel storage requests.

Referring to claim 44: Claim 43's argument applies; furthermore, Lange's fast memory is a cache memory.

Referring to claim 45: Claim 43's argument applies; furthermore, Lange's system controller (figure 1, structure 3) is the communication channel controller receptive to diverse communication protocols (protocols from structures 4, 5, and processors 1-2).

Referring to claim 46: Claim 43's argument applies; furthermore, Gusefski teaches the lock and switch for the cache coherency (column 1, lines 56-64).

18. Claim 49 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gusefski in view of Averill.

Gusefski discloses a plurality of processors and shared resources (figure 2) and the processor has a fast memory (figure 2, structure 18). Gusefski further discloses an exclusive control over the shared resource to only one of the requesting processors (column 6, lines 11-12), and a bus switch unit for scheduling the resource access (column 5, lines 38-40).

Gusefski does not disclose resetting the current processor's cache memory when the current processor is different from the previous processor. Averill teaches that it is known to reset the previous processor's fast memory once the control is switched to the present processor (column 1, lines 34-40).

Hence, it would have been obvious to one having ordinary skill in the computer art to modify Gusefski with Averill's teaching because Averill teaches one on how to resolve the cache coherence problem.

19. Claims 52, 55, 58-59, and 61-63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill.

Referring to claim 52: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11). Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource.

Lange does not reset the memory of the first processor when the resource control is switched out to a second processor; however, Lange resets entity's memory upon gaining the resource control (column 2, lines 7-10, 45-49). Averill teaches that the first processor's memory is reset while the second processor gains the resource control (column 1, lines 34-40).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the teaching of Averill into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control.

Referring to claim 55: Claim 52's argument applies; furthermore, Lange's switch mechanism is a hardware device.

Referring to claims 58-59: Claim 52's argument applies; furthermore, Lange's shared resource is a memory, which is a hardware device.

Referring to claim 61: Claim 52's argument applies; furthermore, the data stored in Lange's shared resource is the software resource.

Referring to claim 62: Claims 52 and 61's arguments apply; furthermore, in order to properly store and retrieve the data, the memory has a particular file system structure, such as the FAT or FAT16, such that, the stored software data source is a data structure.

Referring to claim 63: Claim 52's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is a cache memory.

20. Claim 56 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Tanenbaum.

Referring to claim 56: Claim 52's argument applies; furthermore, neither Lange nor Averill discloses a software switch. Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange and Averill because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

21. Claim 57 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Tanenbaum and Georgiou.

Referring to claim 57: Claims 52 and 56's arguments apply; furthermore, none of Lange, Averill, or Tanenbaum discloses that the switch is a Djikstra primitive. Georgiou teaches implementing distributed arbitration among multiple processors and shared resource with Djikstra primitive. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Georgiou's teaching into Lange, Averill, and Tanenbaum because

Georgiou teaches one to route data with the Djikstra primitive to achieve the shortest path from the source to the destination.

22. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski.

Referring to claim 60: Claim 52's argument applies; neither Lange nore Averill discloses a processor including a communications. Gusefski teaches that it is known to have a processor coupled to a communications channel controller (figure 2, structures 28, 26). Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Gusefski's teaching into Lange and Averill because Gusefski teaches one to couple a central computing unit functioning as a shared channel processor to handle channel storage requests.

23. Claims 64 and 70-73 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski.

Referring to claim 64: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11). Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource.

Lange does not reset the memory of the first processor when the resource control is switched out to a second processor; however, Lange resets entity's memory upon gaining the

resource control (column 2, lines 7-10, 45-49). Averill teaches that the first processor's memory is reset while the second processor gains the resource control (column 1, lines 34-40).

Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource, but Lange does not explicitly disclose the switch mechanism with a lock means to reserve the exclusive control of the resource. Gusefski teaches the switch mechanism coupled to distribute an exclusive control (column 6, lines 11-12).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the teachings of Averill and Gusefski into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control and Gusefski teaches one to lock the shared resource for preventing interrupting data processing.

Referring to claim 70: Claim 64's argument applies, furthermore, the Gusefski's bus switching unit and the shared channel processor are the communication channel controller.

Referring to claim 71: Claim 64's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is cache memory.

Referring to claims 72-73: Claims 64 and 71's argument applies; furthermore, Lange does not specify the cache memory as a primary cache (L1 cache); Gusefski teaches that it is known to use L1 cache (figure 2, structure 18). Every cache type is functioning at the same purpose as to store local information for speedy processing; the type of cache in which each processor uses is inconsequential for the invention as a whole and presents no new or unexpected results, so long as the data on the cache is reset.

24. Claims 67-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Gusefski and Tanenbaum.

Referring to claims 67-68: Claim 64's argument applies; furthermore, Gusefski does not specify whether the lock is a software or hardware. Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange, Averill, and Gusefski because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

Referring to claim 69: Claim 64's argument applies; furthermore, a binary object is generated after the software source code is compiled.

25. Claims 74-77 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill.

Referring to claim 74: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11).

Lange discloses a switch mechanism (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to switch the control to the resource, but Lange does not disclose the switch mechanism coupled to a lock. Gusefski teaches the switch including a lock means for granting exclusive access to only one processor at a time (column 6, lines 11-12).

Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity; however, Lange resets entity's memory upon gaining the resource control (column 2, lines 7-10, 45-49). Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the teachings of Averill and Gusefski into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by resetting previous processor's cache when the second processor gains the resource control and Gusefski teaches one to couple a central computing unit functioning as a shared channel processor to handle channel storage requests.

Referring to claim 75: Claim 74's argument applies; furthermore, Lange's fast memory is a cache memory.

Referring to claim 76: Claim 74's argument applies; furthermore, Lange's system controller (figure 1, structure 3) is the communication channel controller receptive to diverse communication protocols (protocols from structures 4, 5, and processors 1-2). Lange does not disclose a processor coupled to the communication channel. Gusefski discloses a central central computing unit (figure 2, structure 28) coupled to the bus switch unit.

Thus, it would have been obvious to one having ordinary skill in the computer art to adapt the teaching of Gusefski into Lange and Averill because Gusefski teaches one to couple a central computing unit functioning as a shared channel processor to handle the channel storage request.

Referring to claim 77: Claim 74's argument applies; furthermore, Gusefski teaches the lock and switch for the cache coherency (column 1, lines 56-64).

Response to Arguments

26. The claim objects on claims 18, 30-36, and 62 are withdrawn in light of Applicant's explanation.

27. In response to that the claim 1 has been amended to incorporate the limitations of claims 6 and 7 (Remark, page 8, line 2): Although Applicant claims to incorporate the claims 6 and 7 into the claim 1, the claim 1's amended portion does not reflect the claims 6 and 7.

The claims 6 and 7 claim that a memory portion is not reset when the entity is the same entity as the previous one and is selectively reset when the entity is different from the previous one. The claim 1's amended portion claims that a memory portion is reset if the entity does not currently have control of the resource, and is not reset when the entity currently has control of the resource. The claim 1's amended portion disregards the relationship between the current entity and the previous entity; such that the amended claim 1 constitutes new scope of the limitations.

28. In response to the argument on the claim 1's limitation (Remark, page 8): Please see the 103 Rejection above.

29. In response to the argument on the claim 30's limitation (Remark, page 8): Please see the 103 Rejection above.

30. In response to the argument on the claim 33 (Remark, page 9, 2nd paragraph): Please see the 103 Rejection above.

31. In response to the argument on the claims 35-36 (Remark, page 9, 3rd and 4th paragraphs): Please see the 103 Rejection above.

32. In response to the argument on the newly amended claim 37 (Remark, page 9, the last paragraph): Please see the 103 Rejection above on the comparing means and resetting means.

33. In response to the argument on the newly amended claim 42 (Remark, page 10): Please see the 103 Rejection above on the lock means.

34. In response to the argument on the L1 and L2 cache (Remark, page 11, 5th paragraph): The application of L1 and L2 cache is support by DiBrino et al. (U.S. Patent No. 5,581,734, column 1, lines 35-36).

35. In response to the argument on scheduler (Remark, page 12, 3rd paragraph): Barnaby discloses a scheduler as stated in the action 10.

36. In response to the argument on the software switch (Remark, page 12, the last paragraph): Please see the 103 Rejection above.

37. In response to the argument on the Dijikstra (Remark, page 13, 2nd paragraph): Please see the 103 Rejection above for the Dijikstra reference.

38. In response to the argument on channel controller (Remark, page 13, 7th paragraph): The Authoritative Dictionary of IEEE Standards Terms defines the channel as an electric communication, a single path for transmitting electric signals. Since switching mechanisms of Lange, Stevens, and Shah are managing and directing the path for the signals transmitting, the switching mechanisms are the channel controller.

39. In response to the argument on the binary format (Remark, page 14, 4th paragraph): The Authoritative Dictionary of IEEE Standards Terms attributes the binary to the data transmission,

the mathematics of computing, and data management. The mathematics of computing is the executable resource. Furthermore, Operating System Concept by James Peterson and Abraham Silberschatz, as a popular textbook, discloses a binary object to execute (page 5, 4th paragarph).

40. In response to the argument on the newly amended claim 37 (Remark, page 9, the last paragraph): Please see the 103 Rejection above.

Conclusion

41. The prior art made of recorded and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent No. 4,774,625 to Yamanaka: Yamanaka teaches a multiprocessor system with a master processor unit and several slave processor units.

U.S. Patent No. 5,365,526 to Wu: Wu discloses a transaction history tracking for an I/O channel for error correcting purposes.

U.S. Patent No. 4,035,777 to Moreton: Moreton discloses an access control circuit to ensure that only one of the bus masters can access the bus at a time (column 5, lines 55-58).

U.S. Patent No. 3,979,726 to Lange et al.: Lange discloses a selective cache clearing operation while one of entities accesses the resource (column 1, lines 29-52). Lange teaches that it is a performance advantage to selectively clear the portion of the cache with outdated or needless data (column 1, lines 49-52).

Dijkstra's Algorithm by John Morris (http://www.ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/dijkstra.html): Morris teaches that it is

known to apply the Dijkstra's algorithm to solves the problem of finding the shortest path from a source to a destination.

TechEncyclopedia's definition on class: TechEncyclopedia discloses that the class is a user-defined data type that defines a collection of objects.

42. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Justin King whose telephone number is (703) 305-4571. The examiner can normally be reached on Monday through Friday from 9:00 A.M. to 5:00 P.M..

If attempts to reach the examiner by telephones are unsuccessfully, the examiner's supervisor, Mark Reinhart can be reached at (703) 308-3110.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose number is (703)-306-5631.

Justin King
January 13, 2003

Glenn A. Auve
Primary Patent Examiner
Technology Center 2100